RoboBoat 2026: Technical Design Report AGH Solar Boat

Anastasiia Vudvud, Łucja Wuls, Olivier Lichota, Filip Nawalaniec, Aleksander Młynarski, Przemysław Domagała, Michał Zawada, Dawid Stęposz, Mateusz Kalandyk, Mateusz Jaźwiecki, Martyna Czerniak, Julia Czekańska, Patryk Kozłowski, Kacper Kustra, Błażej Dyksiński

AGH University of Kraków, Kraków, Poland

***Abstract*—The autonomous surface vehicle *Barka*, developed by *AGH Solar Boat*, returns to the RoboBoat competition with a strategic focus on localization reliability and deterministic performance. This year's mission readiness is driven by the integration of an RTK base station and a ZED2i camera to eliminate positioning errors, as well as by an upgrade to our modular catamaran platform that adds a dedicated microphone for the new Harbor Alert task.**

The software architecture has transitioned to a scalable Behavior Tree framework, replacing traditional state machines to facilitate independent node testing. Real-time perception now combines YOLOv10 vision detection with LiDAR-based Euclidean clustering, providing a robust environmental model for SLAM-based mapping and A-Star path planning. Validated through MATLAB simulations and extensive trials at Bagry Lake, this integrated approach ensures that Barka handles complex autonomous tasks with high precision and repeatability.

Competition Strategy

Building on the existing system, we integrated a ZED 2i wide-angle camera and an RTK GNSS base station to enhance environmental awareness and eliminate localization drift [1]. Our strategic priority this year is to improve our performance in Navigate the Marina, Emergency Response Sprint, and Harbor Alert. To manage the increased system complexity, we transitioned to a modular behavior tree architecture, allowing for independent testing and deterministic mission execution. All strategies are designed in compliance with the competition handbook [2].

I. Evacuation Route & Return

To address past instabilities caused by GPS-only localization and buoy proximity issues, we now utilize high-precision RTK localization and sensor fusion. The camera reliably detects and tracks buoys, with spatial filtering preventing duplicate detections. The ASV selects the nearest red and green buoys and navigates through a waypoint located at their midpoint. RTK corrections eliminate localization drift, ensuring consistent and accurate passage through the center of the gates.

II. Debris Clearance

In 2025, during the Follow the Path task, we encountered issues with obstacle counting and localization, with some yellow buoys being detected multiple times.

This year, our Debris Clearance approach is based on perception data from the processing layer, which provides tracked buoy positions and classifications to the control system. When gate buoys are visible, the ASV navigates through the channel by generating waypoints at the midpoint of each red–green buoy pair, while storing the followed path.

If the gates are no longer visible, the system switches to a debris scanning behavior, where the vessel reacts to detected objects based on their type. After completing the debris scanning phase, the ASV returns through the channel by following the previously stored path.

III. Emergency Response Sprint

To overcome limited camera FOV and unreliable GPS localization from previous years, we implemented RTK corrections and a structured search pattern to detect the color indicator buoy using vision-based recognition. The ASV then performs a structured search, gradually approaching the expected location of the yellow buoy until it is detected. The sensor fusion of the ZED 2i camera, LiDAR, and IMU ensures accurate buoy localization during the approach. Once confirmed, the ASV executes a clockwise or counterclockwise circling maneuver, continuously refining its trajectory to maintain high speed, smooth motion, and safe clearance, before exiting the course.

IV. Supply Drop

At the previous competition, we completed Object Delivery during the qualifications but did not attempt it in the semi-finals or finals due to other technical challenges.

This year, the Supply Drop task employs a selection algorithm that identifies the nearest target boat. The ASV then generates a two-waypoint approach sequence: an initial standoff point at a configurable distance (default 3.0 m) and a closer approach point at half that distance, both oriented toward the target boat. The tracking system assigns unique IDs to the detected boats and publishes their positions and types to coordinate delivery actions. Once the ASV reaches the target position, it executes the appropriate action sequence, either shooting balls or spraying water, based on the detected boat classification.

V. Navigate the Marina

Recognizing that placard detection alone was insufficient, we developed a geometry-driven strategy using LiDAR-based marina layout modeling for precise alignment [3]. The ASV identifies marina structures as line primitives from LiDAR scans to model slip centers in the global frame. This model generates safe navigation waypoints, including specific entry and exit points.

During transit, the ASV fuses LiDAR and camera depth data to inspect each slip for occupancy, traffic light status, and placard numbers. After evaluating two marina sections, the system selects the optimal slip—one that is vacant, displays a green light, and has the lowest number—and maneuvers to its center, exits, and holds at a designated safe position.

VI. Harbor Alert

The ASV continuously monitors the acoustic environment using an FFT-based audio pipeline. The audio frames are windowed, transformed to the frequency domain, and refined with parabolic interpolation for accurate tone detection.

Detected tones exceeding a magnitude threshold are grouped into blasts and classified as SHORT or LONG based on duration. Blast frequency and count are published to the decision module via ROS 2. When a predefined signal pattern is recognized, the ASV immediately interrupts its current task and navigates to the assigned target zone, while maintaining collision avoidance.

Design Strategy

Our strategy focuses on enhancing a proven modular catamaran platform through high-precision perception and a modular software architecture. To manage complexity and improve reliability, we transitioned from state-machine-based control to behavior trees, enabling scalable and reusable task execution [4]. Key engineering decisions were driven by the need for improved situational awareness and navigation precision, which includes upgrading to wide-angle stereo vision to eliminate blind spots and implementing RTK-GNSS for millimeter-level localization. By refurbishing reliable mechanical components and overhauling the autonomy stack, we achieved a robust system that balances physical stability with decision-making.

I. Mechanical and Hardware Modifications

A. *Hull and Propulsion*

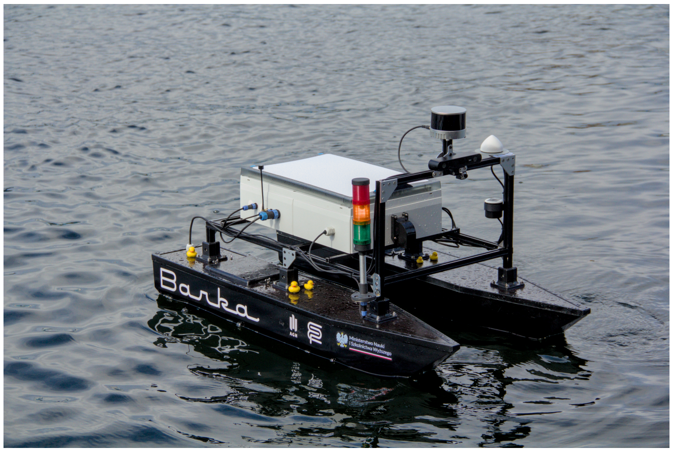


Fig. 1: Image of Barka

The ASV continues to use a modular catamaran hull that supports quick disassembly for transport, maintenance, and air travel. Although the basic hull design remains unchanged, the float modules have been refurbished and refinished for improved durability. With a footprint of 1.3 m×0.9 m, the ASV utilizes two Blue Robotics T200 thrusters located under the rear of each float for propulsion.

B. *Sensor Suite Integration*

To support the new autonomy software, the sensory system was physically expanded.

*1) **Visual**:* For tasks involving multiple visual targets and dynamic maneuvering near gates and obstacles, we upgraded visual perception by replacing the standard Stereolabs ZED2i stereo camera with its wide-angle variant. This expanded the horizontal field of view from 72° to 120°, significantly improving situational awareness without mechanical complexity. The wider sector reduces the likelihood of gates and obstacles leaving the frame during tight maneuvers, enabling stable detection and tracking. Consequently, planning and obstacle avoidance receive consistent perception streams, resulting in smoother trajectories with fewer corrections.

*2) **Audio**:* For the Harbor Alert task, we integrated the Xtrike Me XMC-03 condenser microphone. Its omnidirectional pickup pattern ensures reliable blast detection regardless of vessel orientation. The microphone's USB interface eliminates the need for external preamplifiers or ADCs, reducing EMI and simplifying driver support. Mounted at the front of the aluminum frame and positioned away from the thrusters, it benefits from a lower noise floor, improving signal-to-noise ratio.

*3) **Precision Positioning**:* For reliable and repeatable navigation across the competition course, we implemented a high-precision system using RTK GNSS with INS support [1]. The system consists of a shore-based reference station and an onboard unit connected via long-range radio telemetry. RTK corrections significantly improve the ASV's positioning accuracy and stability, enhancing waypoint tracking, map consistency, and maneuver repeatability. Broadcasting standard RTCM messages over a dedicated link ensures independence from cellular coverage, increasing operational robustness. The shore station computes differential corrections from its fixed position, while the onboard GNSS/INS applies them to resolve carrier-phase ambiguities. This solution integrates seamlessly with onboard drivers and the navigation stack, delivering stable localization for all autonomy modules.

II. Electrical and Communication Systems

A. *Power Management*

The electrical architecture remains largely unchanged from the previous year, as the system has reached a state of operational reliability. The core continues to rely on a custom power-distribution PCB that ensures organized cable management and integrated fuse protection for all electronic components. The Battery Management System remains the primary safety layer, continuously monitoring battery voltage, current, and temperature. It features an automated safety cutoff for the thrusters in critical conditions, alongside an acoustic buzzer and a 360° LED status light post to provide immediate visual feedback on the ASV's operational mode.

B. *Internal Network*

To comply with the competition communication rules, we implemented a local Wi-Fi network using two Ubiquiti Bullet M2 devices configured in bridge mode. The shore unit operates as an access point with a sector antenna, while the ASV unit connects in station mode using an omnidirectional antenna. Their high transmission power and dedicated antennas ensure stable long-range communication. Onboard, the Wi-Fi link is integrated with the ASV LAN, connecting the Jetson computer and LiDAR via a 1 Gbps switch. To maintain proper bridge operation, only the two Bullet devices are included in the bridge network, while a separate shore access point is used for the crew's devices.

III. Software Architecture

A. *Layer 0: Hardware Drivers*

Layer 0 (L0) serves as the primary interface for all hardware communication and contains low-level messaging, CAN Bus interactions, and motor control routines. This layer uses the ZED wrapper, the Hesai ROS wrapper, and a custom CAN adapter to manage data from the hardware components. By incorporating GNSS data with RTK corrections, the system attains the high-precision positioning necessary for accurate navigation and mapping.

A central component of L0 is the custom CAN adapter, which implements the essential communication bridge between the high-level software stack and the vehicle's hardware. Beyond thruster control, this layer contains the logic required to handle peripherals and to format and send the appropriate CAN messages to operate the water pump and the custom ball gun. This ensures that all sensors and actuators are synchronized with high-level control logic, providing a robust and reliable foundation for physical execution.

B. Layer 1: Object Detection

Following the recent architectural refinement, the system formerly known as the Data Processing Layer has been streamlined and renamed Object Detection. Layer 1 (L1) now focuses exclusively on real-time perception, data fusion, and object tracking.

This layer provides the critical perception capabilities of the vessel, processing vision and point cloud data to identify and track obstacles in real time. L1 serves as a dedicated detection engine that ensures high-fidelity situational awareness.

*1) **Autolabeling & Augmentation**:* To streamline model training, we utilize a partial auto-labeling workflow. A subset of our dataset is manually annotated to train an initial model, which then auto-labels the remaining raw data. This process reduces the remaining manual labor to a review-only phase, significantly accelerating the preparation of the dataset. We also apply extensive data augmentation to simulate conditions such as intense sunlight and glare, ensuring the model remains robust across varying maritime environments.

*2) **Optimized Neural Network Inference**:* For real-time vision, we use YOLOv10 [5], which offers superior accuracy and speed relative to its previous iterations. To achieve the low latency required for autonomous operations, we implement quantization, reducing weight precision from 32-bit to 16-bit with negligible accuracy loss. Using frameworks such as CUDA and TensorRT [6], we optimize parallel computing on the GPU. The workflow includes a 10-step warm-up to initialize GPU resources, followed by image preprocessing (resizing and normalization), model inference, and conversion of outputs into ROS2-compatible message formats.

*3) **Clusterization & LiDAR Processing**:* In parallel with the vision pipeline, we process LiDAR point clouds using the Euclidean Clustering algorithm implemented with the Point Cloud Library (PCL) [7]. This method segments objects based on their spatial proximity, providing precise distance measurements. This year, we enhanced this process by incorporating camera-derived point clouds from the ZED2i, yielding a denser point set that improves the initial object estimates and the reliability of the subsequent fusion.

*4) **Detection Fusion & Tracking**:* The core of L1 is the Detection Fusion system, which synchronizes data from both the vision and cluster-based detectors using precise timestamps. The system evaluates the angular positions of detections from both sources to determine if they represent the same physical object. Once fused, the system integrates LiDAR and camera data with the Xsens IMU/GNSS inputs to calculate global coordinates, relative to the vessel's starting position.

Each detection is processed through a tracking algorithm that compares new coordinates with the locations of previously identified objects. Based on a defined distance threshold, objects are either assigned to a new ID or matched to an existing one. This persistent tracking allows the system to distinguish between various buoys and obstacles, maintaining a stable environmental model for the higher-level SLAM and planning layers.

C. Layer 2: Data Processing and SLAM

*1) **Layer Components and Signal Processing**:* This layer (L2) performs the calculations required for waypoint navigation and detecting complex structures from LiDAR, such as marinas or docks. The SLAM mapper and planner nodes are located here, together with several supporting components. The Odometry node is responsible for running a Kalman filter to estimate the vessel position using IMU acceleration data and GNSS measurements with RTK corrections. The pointcloud_to_laserscan node converts the 3D LiDAR point cloud into a 2D laser scan, so that it can be processed by the mapping and the segment_scan nodes, where the latter is used to aggregate points from the laser scan into straight lines to detect docking slips or a marina by their dedicated detectors. To precisely follow the calculated path, we use our recently implemented controller system with a physical model of our boat.

*2) **SLAM and Path Planning Strategy**:* To complete tasks and navigate precisely, we need a reliable tool to keep us oriented in the surroundings. The Simultaneous Localization and Mapping system turned out to fit our architecture perfectly [8], specifically when it comes to the Mapper and Planner functionality. Firstly, the data from LiDAR is processed into a local map and synchronized with position information from odometry in order to form and update a global map. The costmap is then generated from the global map. Concurrently, the fused detections from Layer 0 are compared with those already seen and combined to keep track of objects on course. Finally, as soon as Layer 3 establishes the goal, the planning process takes place. We use the A^* algorithm with a modified Euclidean heuristic run on the costmap to find a safe and efficient path to the goal [9]. The plan is updated as the vessel makes its way to the waypoint to ensure no collisions with new objects.

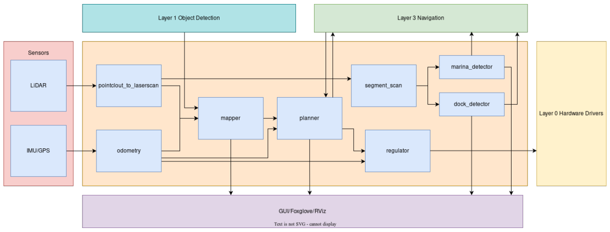


Fig. 2: Software architecture

*3) **Motion Control**:* The controller node processes sensor data such as position, velocity, and heading, and compares them with reference values derived from the planned trajectory. Based on the resulting error, it generates control commands for propulsion and steering to ensure stable motion and accurate path tracking. The controller's objective is to minimize position and orientation errors while maintaining smooth and safe maneuvers. The node must also respond to external disturbances, such as wind, waves, and varying load conditions, compensating for their influence on the ASV's motion and, as a result, the vessel can autonomously perform navigation tasks, maintaining the desired course and speed even under changing environmental conditions.

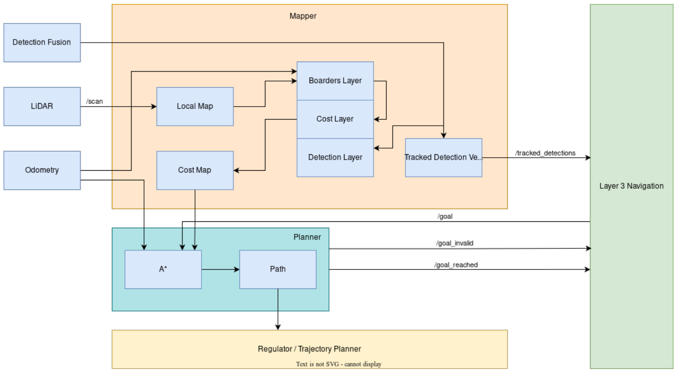


Fig. 3: High-level steering architecture

D. *Layer 3: Navigation*

The previous control architecture was based on a state machine that managed the execution flow of the algorithms implemented for specific tasks. This approach was effective during the early development stages that involved a small number of well-defined missions. However, as the system grew and the number and complexity of supported tasks expanded, the limitations of a state-machine-based approach became apparent. To improve scalability and maintainability, behavior trees were introduced [4], which are easier to extend and reorganize, allowing new behaviors to be added without redesigning the entire control structure. Rather than implementing whole tasks, individual tree nodes encapsulate smaller functionalities. This modular approach enables the reuse of nodes across multiple sequences, reducing code duplication, and simplifies testing and debugging, as nodes can be developed and validated independently before integration into the central decision-making layer.

Layer 3 (L3) is responsible for making decisions, waypoint generation, and task execution. Its primary node, basic_controller, handles behavior tree ticking and coordinates a hierarchy of control and utility subnodes. These custom subnodes implement actions such as spinning, breaking, steering, waypoint following, and perception-driven searches for buoys, boats, and docks. In short, L3 receives data about known objects from L2 and then computes task-specific waypoints to accomplish the assigned objectives [10].

E. *Human-Machine Interface (HMI)*

To provide a clear and reliable view of the system state, we use Foxglove as a diagnostic and monitoring platform integrated with our ROS 2 architecture [11]. Its GUI serves as the primary operator interface for streaming high-bandwidth ROS topics from the ASV. Foxglove_bridge establishes a wireless link between the onboard computer and the operator station, enabling real-time visualization of camera feeds, point clouds, and maps generated by SLAM and planning modules. This level of feedback is essential during water trials, as it allows engineers to observe system behavior, diagnose issues in real time, and validate performance efficiently and safely.

Testing Strategy

Testing is a crucial part of the development process and a key indicator of progress. We divide our tests into three parts: simulation, dry, and on-water tests. In addition, this year we also employed MATLAB-based simulations to validate our controller.

I. Simulation tests

The basis for high-level software verification is a VRX world executed in Gazebo simulation environment [12], [13]. Each new functionality undergoes thorough validation in simulation before deployment on the physical platform, maximizing effective use of on-water testing time. The simulation framework enables systematic refinement of algorithm logic and verification of perception and planning pipelines.

MATLAB-based Simulink is used to validate the motion control algorithm. The performed simulations demonstrate the controller's ability to minimize position and orientation errors relative to the planned trajectory and to compensate for external disturbances such as wind and waves. The results confirm the ASV's capability to autonomously maintain the desired course and speed under varying conditions without requiring immediate on-water testing.

II. Dry tests

To validate hardware robustness, we conduct dry tests in our workshop, performing component-level checks to ensure key elements, such as the stereo camera, ball shooter, or LiDAR, function correctly before full system activation. The nozzle is tested by connecting it to the boat and manually issuing commands to start and stop water flow, and similar procedure is applied to the ball shooter. After the monitored responses are sufficient, the next stage involves activating the entire boat, which begins with verifying correct signal transmission from the radio to the thrusters. Then, the main autonomous onboard component, Jetson AGX Orin, is activated and checked for proper communication with all sensors and actuators. Finally, we validate core software functionalities, including object detection and waypoint navigation, while observing transmitted data and thruster outputs. These dry tests provide a quick and convenient way to evaluate features that do not require water.

III. Water tests

We conduct on-water tests on two lakes in Kraków: Bagry and Zakrzówek. We already had had buoys for the Navigation Channel and Speed Challenge, as well as placards for the Object Delivery task. This year, we built a dock from extruded polystyrene foam to enable Docking task validation. This setup allows us to exercise the full task set on water. Additionally, we maintain a labeled dataset coveting all of these object types used to train our object detection models. This approach ensures that our testing scenarios closely mirror competition conditions and enable iterative refinement and optimization of Barka's performance in a controlled setting.

Acknowledgments

We extend our heartfelt gratitude to AGH University of Kraków for their invaluable support and resources throughout the development of this project. We are especially grateful to our academic advisors, Krzysztof Sornek and Wojciech Sajdak, for their expertise, guidance, and dedication.

Our deepest thanks also go to TME, Xsens, and ABB for their crucial support in design and funding. We thank MINE AGH for filming our promotional video and providing media materials and also appreciate the HORN sailing club for enabling on-water tests and the VRX developers for their dynamic simulation environment.

A special gratitude goes to our team members: Łucja Wuls, Michał Zawada, Ewa Nowicka and Przemysław Domagała, for preparing the application for the Ministry of Science and Higher Education scholarship, without which our participation in this competition would have been much more challenging.

Finally, we extend our deepest gratitude to every member of our team for their dedication, time, and passion that made this project a reality.

References

[1] P. J. G. Teunissen and O. Montenbruck, Eds., *Springer Handbook of Global Navigation Satellite Systems*. Springer International Publishing, 2017.

[2] Robonation Inc., "2026 team handbook," 2025, accessed: 2025. [Online]. Available: robonation.gitbook.io/roboboat-resources

[3] K. Miądlicki, M. Pajor, and M. Saków, "Ground plane estimation from sparse lidar data for loader crane sensor fusion system," in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2017, pp. 717–722.

[4] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI: An Introduction*. CRC Press, 2018.

[5] A. Wang *et al.*, "Yolov10: Real-time end-to-end object detection," *arXiv preprint arXiv:2405.14458*, 2024.

[6] Ultralytics, "Yolov10 - ultralytics yolo docs," 2024, [Online]. [Online]. Available: https://docs.ultralytics.com/models/yolov10/

[7] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4.

[8] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[9] D. Bruijnen, J. van Helvoort, and R. van de Molengraft, "Realtime motion path generation using subtargets in a changing environment," in *2006 American Control Conference*, 2006, p. 6 pp.

[10] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[11] Foxglove Technologies Inc., "Foxglove documentation," 2024. [Online]. Available: https://docs.foxglove.dev/docs

[12] B. Bingham *et al.*, "Toward maritime robotic simulation in gazebo," in *OCEANS 2019 MTS/IEEE SEATTLE*, 2019, pp. 1–10.

[13] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2004, pp. 2149–2154.

Appendix A
Test Plan & Results

A. Scope

We conduct monthly tests of our ASV, starting with trivial functionalities and gradually progressing to more advanced ones. Basic checks include verifying that the electrical system, detection fusion, and waypoint navigation function as expected. As competition approaches, we focus on testing more advanced functionalities, such as task-specific behaviors.

B. Schedule

The tests are organized in a way that allows us to validate the fundamental capabilities first and test the new features later. After assessing the functionality of our ASV during Njord - The Autonomous Ship Challenge 2025, our planned schedule is as below:

- **November:** Depth from ZED 2i camera, SLAM mapper and planner
- **December:** ASV controller, behavior trees for basic functionalities, such as swimming through a gate made out of two buoys
- **January:** RTK base station, behavior trees for Navigate the Marina, Emergency Response Sprint, Evacuation, and Debris Clearance
- **February:** Hardware & behaviour tree for Harbor Alert, ball launcher

C. Environment

Our simulation is built on a world originally created for the VRX competition by using either custom-made or VRX models and assets. Each task from the RoboBoat 2026 Team Handbook, other than the Harbor Alert one, was recreated allowing for developing algorithms that require continuous testing.

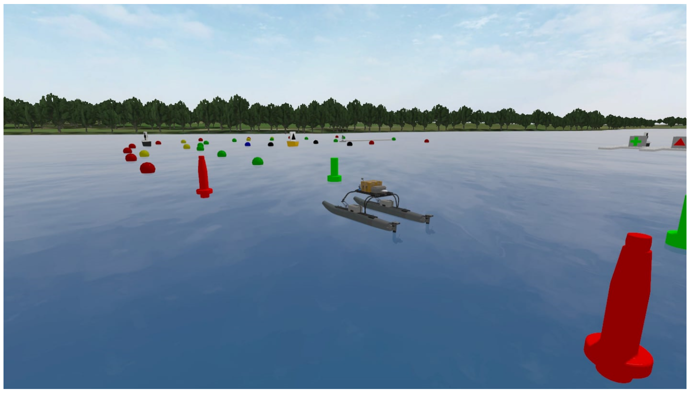


Fig. 4: VRX simulation environment of the RoboBoat 2026 course

In MATLAB, we are able to see how the steering algorithm controls the ASV, which allows us to perfect the control parameters and ensure the vessel minimizes trajectory errors and executes smooth maneuvers. In Figure 5, the dash-dotted green and purple lines show set left and right thruster commands, while the red and blue solid lines are speeds on X and Y axes, respectively.

For on-water testing we utilize Bagry Lake in Kraków, specifically the harbor of the HORN Sailing Club. It provides a safe environment, often used by beginner sailors, and allows us to position buoys and other visual marks as needed. Although it is our usual testing ground, this lake tends to freeze during colder months, forcing our team to move to Zakrzówek Lake.

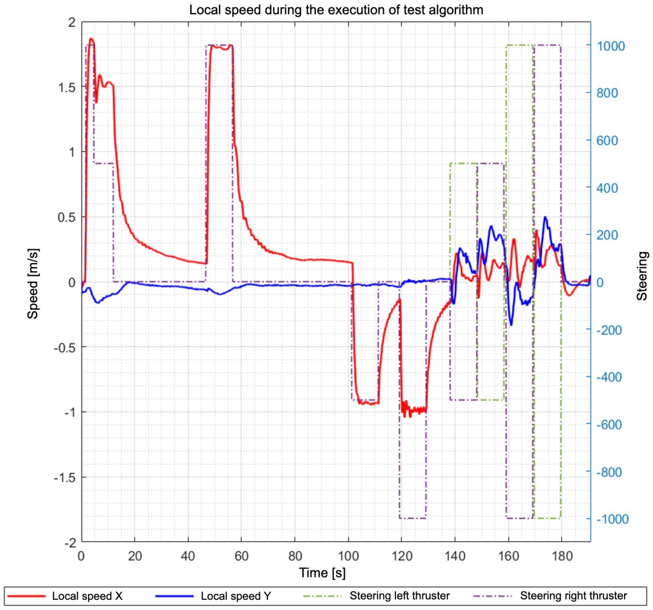
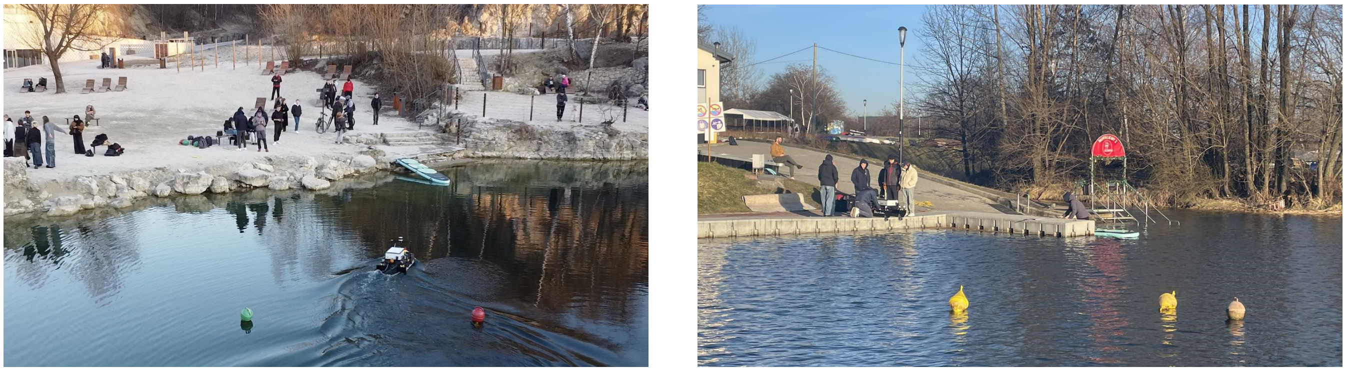


Fig. 5: Left and right thruster commands against X and Y speeds



(a) Zakrzówek Lake

(b) Sailing club HORN in Krakow

Fig. 6: Water testing environments

D. *Risk Management*

We regularly conduct on-water tests not only for our ASV Barka, but also our solar-powered racing boats, which ensures that we are well prepared when dealing with potential hazards including slips and accidental immersion when setting up or retrieving the equipment from a lake. To mitigate these risks we use paddleboards or kayaks when modifying the course, require lifejackets for everyone involved, and ensure a safety briefing for each member of our team.

E. *Testing results*

Recent tests confirmed correct operation of core hardware, i.e. the nozzle, ball launcher, and microphone all met functional expectations. The sensor fusion between the stereo camera and LiDAR performs reliably, and the mapper and planner nodes generate consistent global maps and costmaps, which allowed us to record real-time data that will support further algorithm refinement. Water trials also demonstrated that the controller computes appropriate thruster commands to follow waypoint trajectories with the precision required for task execution. Despite harsh winter weather in Kraków, we managed to successfully begin testing half of the tasks on water and most of them, save for the Harbor Alert task, in the Gazebo environment simulation.

Over the next month we will complete remaining functionality tests identified in the schedule. Additionally, we plan to conduct further validation during on-site trials in Sarasota prior to the competition deployment.